Binary output A/D converter for microprocessors

by Peter Bradshaw,
Intersil Inc.

The Intersil range of A/D converters now includes two-chip combinations for high resolution conversion and with coded outputs. Characteristics of the devices are described with several examples of application. They are expected to be available in the U.K. in May, this year.

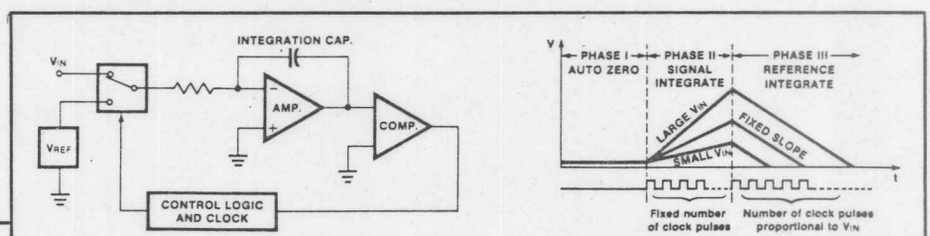
The ICL7104, combined with the ICL8052 or ICL8052LN, forms a member of Intersil's high performance A/D converter family. The 16-bit version, the ICL7104-16, performs the analogue switching and digital function for a 16-bit binary A/D converter, with full three-state input, u.a.r.t. handshake capability, and other outputs for a wide range of output interfacing. The ICL7104-14 and ICL7104-12 are 14- and 12-bit versions respectively. The analogue section provides precise auto-zero, auto-polarity (including ±0 null indication), single reference operation, very high input impedance, true input integration over a constant period for maximum e.m.i. rejection, fully ratiometric operation, over-range indication, and a medium quality built-in reference. The chip pair also offers optional input buffer gain for high sensitivity applications, a built-in clock oscillator, and output signals for providing an external auto-zero capability in pre-conditioning circuitry, synchronising external multiplexers, etc. The basic schematic connections are shown in Fig. 1.

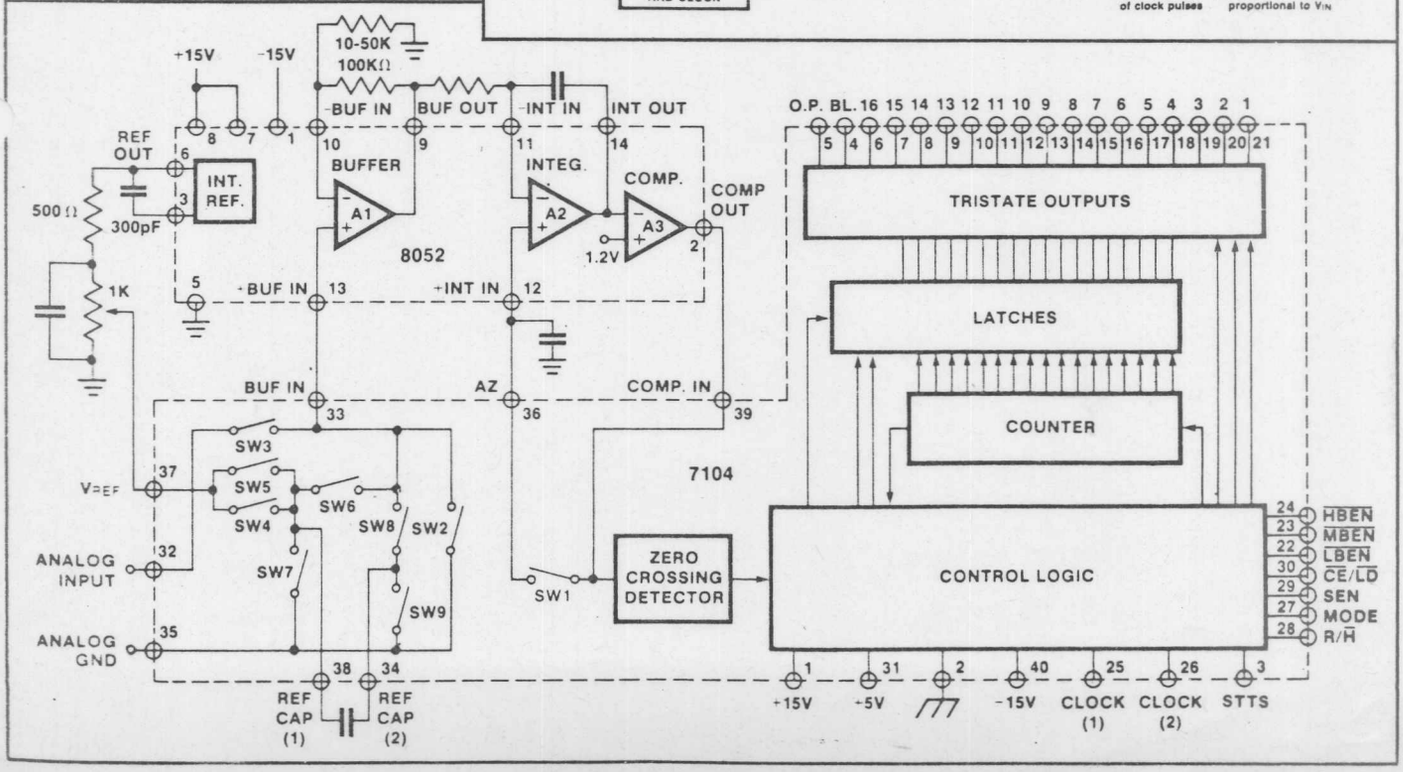
The chip pair operates as a dual-slope integrating converter. The conversion takes place in three stages, each with its own configuration. In the first, or auto-zero phase (this is also the "idle" condition), the converter self-corrects for all the offset voltages in the buffer, integrator, and comparator. During the second, or input integrate phase, the converter integrates the input signal for a fixed time (2^{15} clock pulses for the 16-bit part, 2^{13} for the 14-, and 2^{11} for the 12-bit device). The converter then determines the (average) polarity of the input, and during the third, or deintegrate (alias reference integrate) phase, integrates the reference voltage in the opposite polarity, until the circuit returns to the initial condition. This point is known as the zero-crossing, and terminates the conversion process. The time (number of clock pulses) required to reach zero-crossing is proportional to the ratio of the input signal to the reference. Fig. 2 shows the basic waveforms of the integrator.

This article will first cover the digital interface of the ICL8052-7104 chip pair to digital systems of various kinds, ▶40

Fig. 1: (below) Type 8052A/7104 twelve to sixteen bit analogue to digital converter.

Fig. 2: (right) Simplified dual-slope converter and the three phases of dual-slope conversion.

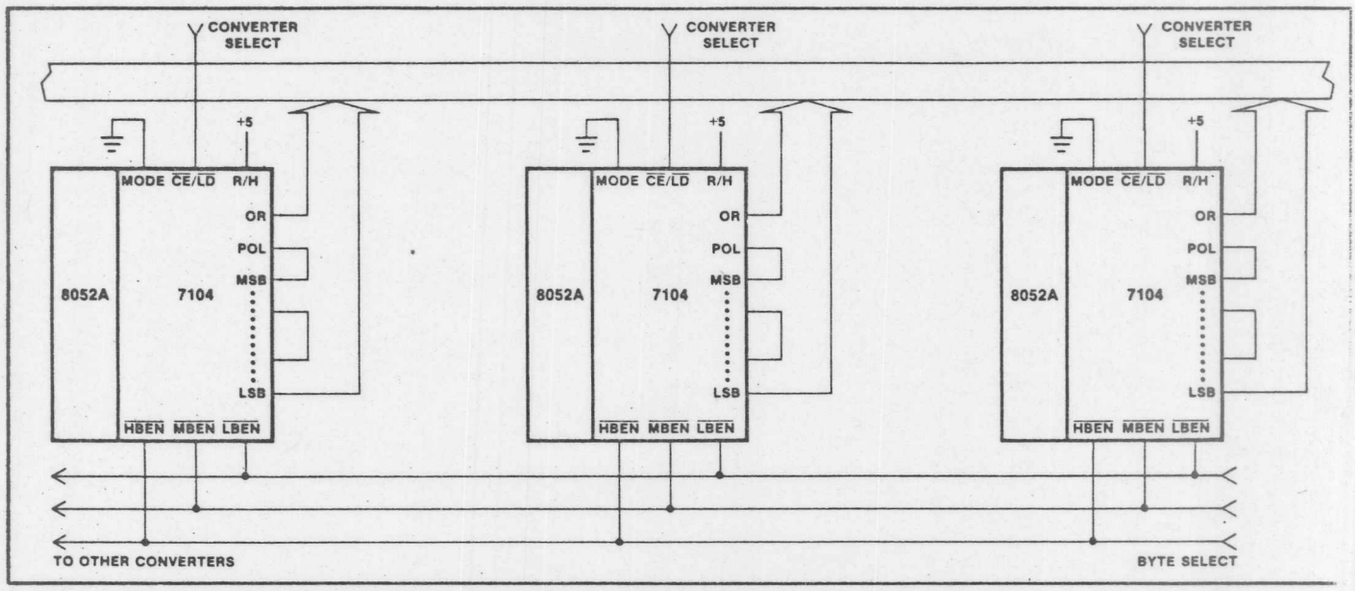




38◀ including microprocessors, using the three state output capabilities, followed by the handshake system built into the 7104. Finally, some (mainly) analogue techniques to enhance the system performance in certain applications are covered.

Fig. 3: Tristating several 7104s to a small bus.

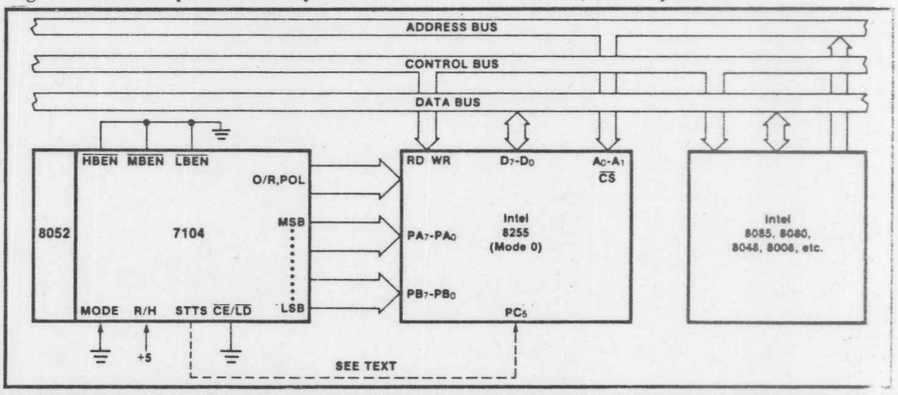
The output format of the ICL7104 includes a full handshake capability, which is discussed later. Here we will be concerned only with the "normal" three state output lines. To disable the handshake circuitry, the *mode* pin (pin 27) should be tied low. In this mode, the most useful output-timing signal is the *status* (STTS) line (pin 3), which goes high at the beginning of the signal integrate phase. When zero crossing occurs (or overload detection), new data is latched on the next clock pulse, and one clock pulse later, the STTS line goes low. Thus, the new data is stable on this transition. The Run/Hold pin (R/H) (pin 28) is also useful for controlling conversions. If it is high, conversions will be performed continuously, while if it ▶44



404 is low, the current conversion will be completed, but no others will start until it goes high again. There are 18 data output lines (16 and 14 on the 14-bit and 12-bit versions), including the polarity and over-range lines. These lines are grouped in sets of no more than eight for three-stated enable purposes, in the format shown in Fig. 3, under the control of the byte and chip disable lines shown. If all four (three for 7104-14 and -12) disable lines are tied low, all the data output lines will be asserted full time, thus giving a latched parallel output. For a three-state parallel output, the three byte disable lines should be tied low, and the chip disable line will act as a normal three-state control line This technique assumes the use of an 18 bit wide bus, fairly common among mini-computers and larger computers, but still rare among microprocessors (note that "extra" bits can sometimes be sensed as condition flags, etc.). For small words, the bit groups can be enabled individually or in pairs, by tying the chip disable line low, and using the byte disable lines either individually or in any combination as three-state control lines. Several devices can be three-stated to one bus by the technique suggested in Fig. 3. comparable to row and column selection in memory arrays.

Some practical interface circuits utilising the parallel and three-state output capabilities of the ICL7104 are shown in Fig. 7 through 13. Figure 4 shows a straightforward application to the Intel MCS-48, 80, and 85 systems via an 8255 programmable peripheral interface, using full-time parallel output. The I/O ports of an 8155 can be used in the same way. This interface can be used in a read-anytime mode, although there can be timing problems here, since a read performed as new data is being latched in the ICL7104 may lead to scrambled data. One way to overcome this problem is to read back the STTS line as well, and if it is high, read the data again after a delay exceeding one (converter) clock cycle. If STTS is now low, the second reading is correct, if it is still high, the first reading was correct (note that data never changes when STTS is low, and it goes low one clock cycle after data update occurs). Alternatively, the problem is completely avoided by using a read-after-update mode, as shown in Fig. 5. Here the high to low transition of STTS triggers a read data operation through the MCS-8 interrupt system. This application also shows the R/H pin being used to initiate conversions under software control.

Fig. 4: Full time parallel interface 8052-7104 to MCS-48, 80/85 families.

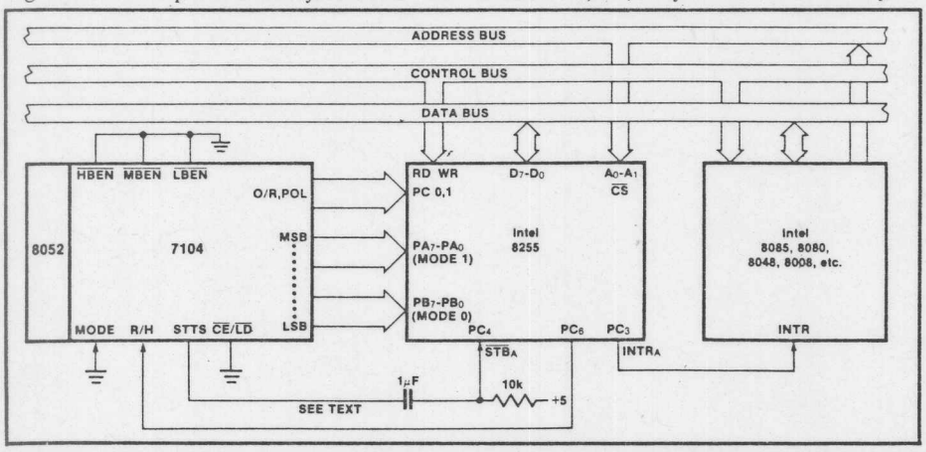


If continuous conversions are desired, R/H may be held high, and if the maximum possible conversion rate is desired, R/H may be tied to clock out.

A similar interface to the Motorola MC6800 system is shown in Fig. 6. Since the maximum input-port count here is only 16, while the 16-bit ICL7104 has 18 outputs, control register A is used to input the two extra bits. The high to low transition of the STTS pin enables the two high bits, clocking the two interrupt flags in control register A if they are negative. A pullup resistor is needed on CA1, though CA2 has one internally. The same transition causes an interrupt via the control register B CB1 line. It is important to ensure that the software interrupt routine reads control register A before reading data port A, since the latter operation will clear the interrupt flags. Note that CB2 controls the R/H pin through control register B, allowing software initiation of conversions in this system also. Naturally, the 14- and 12-bit versions of the ICL7104 avoid this problem since 16 or fewer bits need to be read back. Since the MOS Technology MCS650X microprocessors are bus-compatible with the MC6800, the same circuit can be used. Figure 7 shows an interface to the Intersil IM6100 microprocessor family through the IM6101 PIE device. Here the data is read back in a 10-bit and an eight-bit word, directly from the 7104 on to the 12-bit data bus. Again, the high to low transition of the STTS line triggers an interrupt. This leads to a software routine which controls the two read operations. As before, the R/H pin is shown under software control, though the options already mentioned are equally acceptable, depending on system needs.

These interrupt-fed systems essentially use an external handshake operation, under software control. An interesting variation, using the simultaneous direct memory access capability of the IM6100 family, is shown in Fig. 8. The IM6102 MEDIC allows d.m.a. during bus-idle processor cycles, so the transfer takes no extra time. The current address and extended current address registers of the IM6102 control the memory location to which the data will be sent, and the STTS output of the ICL7104 allows data transfer only when the converter is not updating information. The ECA register is used to ▶ 48

Fig. 5: Full time parallel interface 8052-7104 to MCS-48, 80, 85 families with interrupt.



45 ◀ drive the byte select lines (CA should be set to 7777, and WC to 2) and the user pulse controls chip disable CE/LD. A more fully loaded system can use address latches for CA. A d.m.a. system can also be set up on the MCS-8 system using the d.m.a. controller, 8257, and the three-state outputs of the ICL7104.

It is possible using the three-state output capability, to connect the ICL7104 directly on to many microprocessor busses. An example of this is shown in Fig. 9. It is necessary to consider the system timing in this kind of application, and careful study should be made of the required set-up times from the microprocessor data sheets. Generally this type of circuit is only favoured if the memory peripheral address density is low, so that simple redundant address decoding can be used. Interrupt handling can require multiple external components also, and use of an interface device is normally advisable if this is needed.

Handshake mode interface

Entry into the handshake mode will occur if either of two conditions are fulfilled; first, if new data is latched (i.e. a conversion is completed) while *mode* pin (27) is high, in which case entry occurs at the end of the latch cycle; or secondly, if the *mode* pin goes from low to high, when entry will occur immediately (if new data is being latched, entry is delayed to the end of the latch cycle). While in the handshake mode, data latching is inhibited, and the *mode* pin is ignored. (Note that conversion cycles will continue in the normal manner.) This allows versatile initiation of handshake operation without danger of false data generation; if the *mode* pin is held high, every conversion (other than those completed during handshake operations) will start a new handshake operation, while if the *mode* pin is pulsed high, handshake operations can be obtained "on demand".

During handshake operations, the various "disable" pins become output pins, generating signals used for the handshake operation. The *send enable* pin (SEN) (pin 29) is used as an indication of the ability of the external device to receive data. The condition of the line is sensed once every clock pulse, and if it is high, the next (or first) byte is enabled on the next rising CL1 (pin 25) clock edge, the corresponding byte disable line goes low, and the *chip disable/load* line (pin 30) (CE/LD) goes low for one full clock pulse only, returning high.

On the next falling CL1 clock pulse edge, if SEN remains high, or after it goes high again, the byte output lines will be put in the high impedance state (or three-stated off). One half pulse later, the byte disable pin will be cleared high, and (unless finished) the CE/LD and the next byte disable pin will go low. This will continue until all three (two in the case of 12- and 14-bit devices) bytes have been sent. The bytes are individually put into the low impedance state i.e.: three-stated on during most of the time that their byte disable pin is low. When receipt of the last byte has been acknowledged by a high SEN, the handshake mode will be cleared, re-enabling data latching from conversions, and recognising the condition of the *mode* pin again. The byte and chip disables will be three-stated off, if the *mode* pin is low, but held high by their (weak) pull-ups. These timing relationships are illustrated Fig. 10..

This configuration allows ready interf with a wide variety of external device instance, external latches can be clocked the rising edge of CE/LD, and the ▶

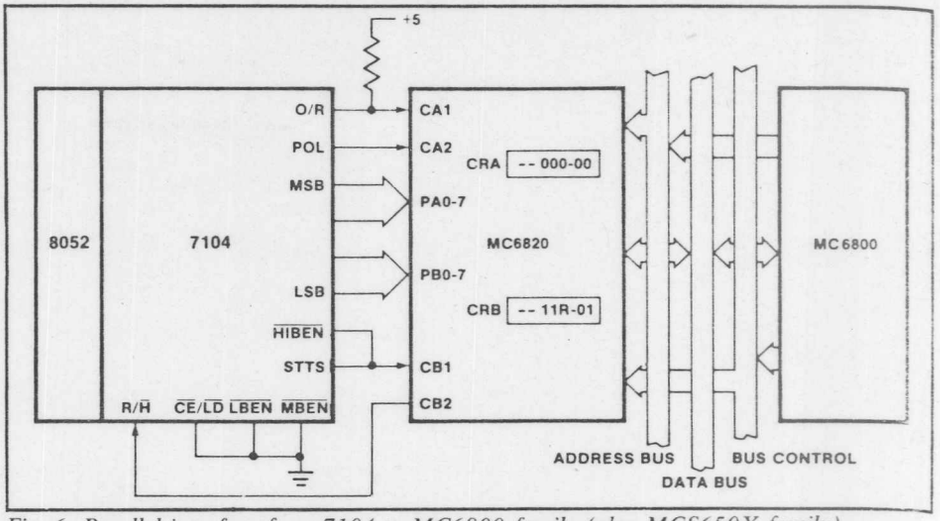


Fig. 6: Parallel interface from 7104 to MC6800 family (also MCS650X family).

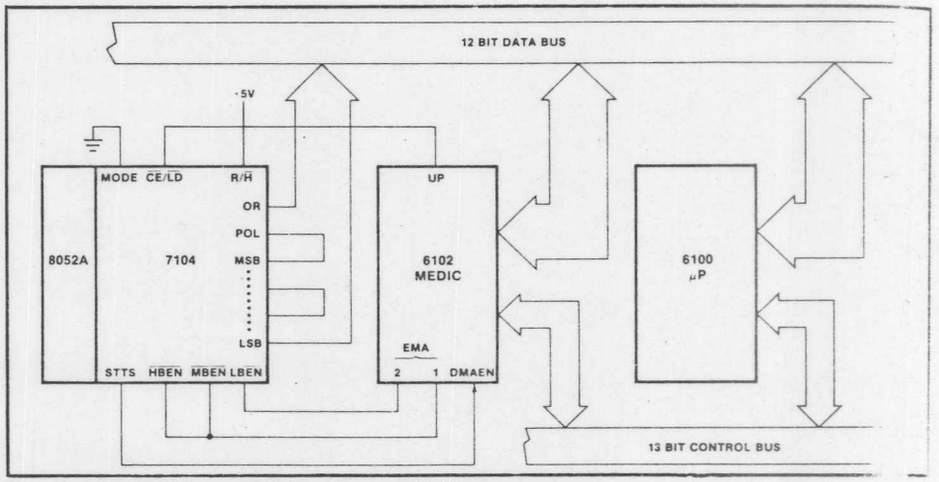


Fig. 7: 8052/7104 parallel interface with 6100 microprocessor.

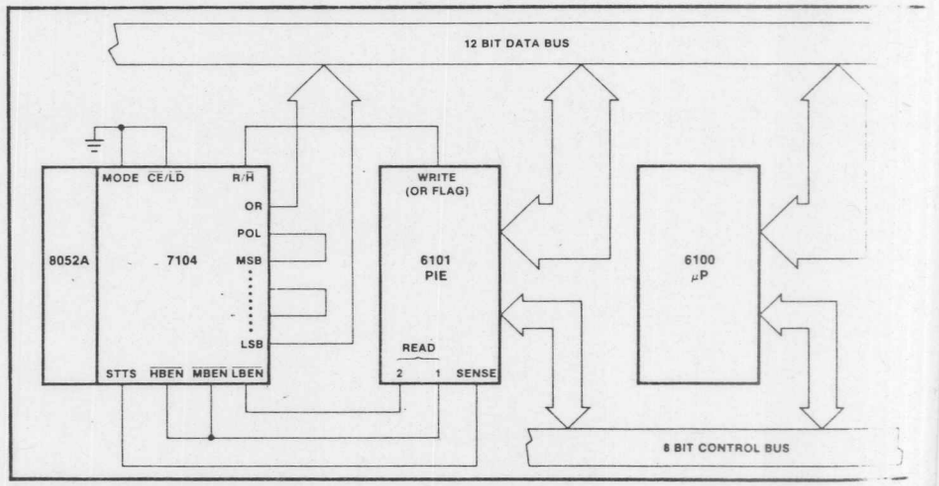


Fig. 8: 8052/7104 parallel interface with 6100 microproprocessor using d.m.a.

48 ◀byte disables can be used to drive either load enables, or provide data identification flags. More usefully, the handshake mode can be used to interface with an eight-bit microprocessor of the MCS-8 group (eg. 8048, 8080, 8085, etc.) as shown in Fig. 11. The handshake operation with the 8255 programmable peripheral interface is controlled by inverting its input buffer full flag to drive the *send enable* pin, and driving its strobe with the CE/LD line. The internal control register of the p.p.i. should be set in mode 1 for the port used. If the 7104 is in handshake mode, and the 8255 i.b.f. flag is low, the next word will be presented to the chosen port, and strobed. The strobe will cause i.b.f. to rise ▶ 179

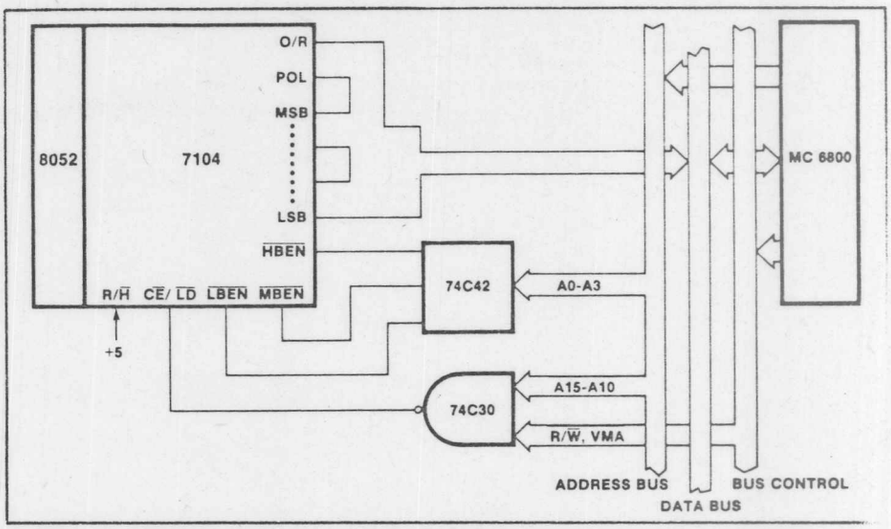
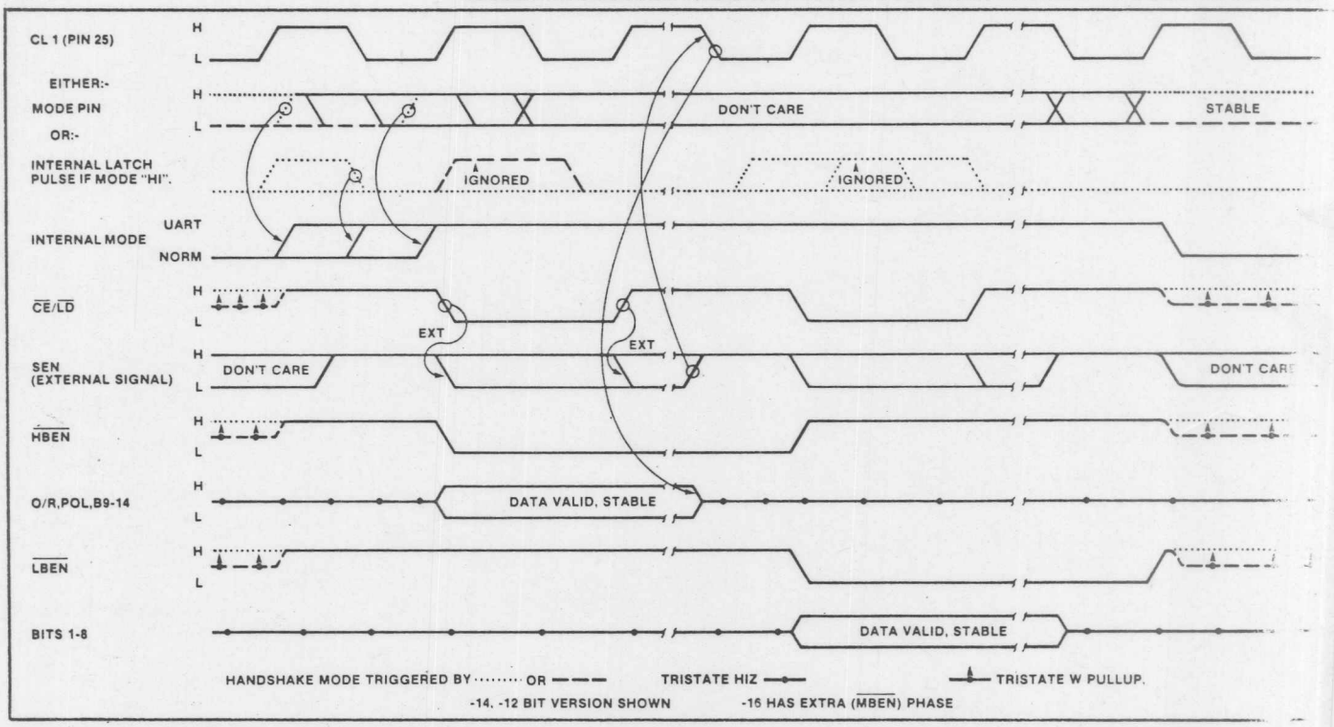


Fig. 9: Direct 8052/7104 to MC6800 microprocessor interface.

Fig. 10: Timing relationships in handshake mode.



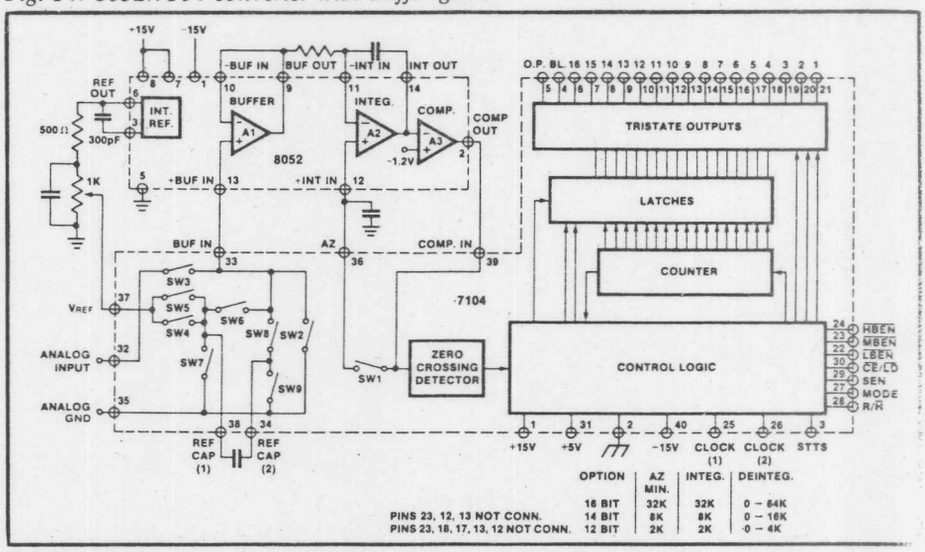
179 ◀ extending the benefits of auto-zero operation to preamplifiers, etc., to cover specialised signal processing or sensitivity enhancement by 10-100 to 1; and a special interconnection to allow the maximum rate of conversion with lower-valued inputs.

One of the significant contributions to the effective input noise voltage of a dual slope integrator is the so called auto-zero noise. At the end of the auto-zero interval, the instantaneous noise voltage on the auto-zero capacitor is stored, and subtracts from the input voltage while adding to the reference voltage during the next cycle. Although the open loop band width of the auto-zero loop is not wide, the gain from the input is very high, and the resulting closed loop band width to buffer noise is fairly wide. The result is that this noise voltage effectively is somewhat greater than the input noise voltage of the buffer itself during integration. By introducing some voltage gain into the buffer, the effect of the auto-zero noise (referred to the input) can be reduced to the level of the inherent buffer noise. This generally occurs with a buffer gain of between 3 and 10. Further increase in buffer gain merely increases the total offset to be handled by the auto-zero loop, and reduces the available buffer and integrator swings, without improving the noise performance of the system. The circuit recommended for doing this with the ICL8052 (LN)/ICL7104 is shown in Fig. 14. With careful layout, the circuit shown can achieve effective input noise voltages of the order of 1-2μV, allowing full 16-bit use with full scale inputs of as low as 150mV.

The R/H pin (pin 28) can be used in two basic modes. If it is held high, the ICL7104-16 will perform a complete conversion cycle in 131k clock counts (strictly 2^{17}), regardless of the result value.

If, however, the R/H pin ever goes low between the time of the zero-crossing and the end of a full count reference integrate phase, that phase is immediately terminated. If it is then held low, the 7104 will ensure a minimum auto-zero count and then wait in auto-zero until the R/H pin goes high. On the other hand, if it goes high immediately subsequent to this minimal auto-zero count, the 7104 will start the next conversion after the least permissible

Fig. 14: 8052/7104 converter with buffer gain.



50 ◀ locking the three-stated byte on. The p.p.i. will cause a program interrupt in the MCS-8 system, which will result (after the appropriate program steps have been executed) in a "read" operation. The byte will be read, and the i.b.f. reset low. This will cause the current byte disable to be dropped, and the next (if any) selected, strobed, etc., as before. The interface circuit as shown has the *mode* pin tied to a control line on the p.p.i. If this bit is set always high (or mode is tied high separately), every conversion will be fed into the system (provided that the three interrupt sequences take less time than one conversion) as three 8-bit bytes; if this bit is normally left low, setting it high will cause a data transmission on demand. The interrupt routine can be used to reset the bit, if desired. Note also that the R/H pin is also shown tied to a control bit so that conversions can be performed either continuously or on demand under software control. Note that one port is not used here, and can service another peripheral device. The same arrangement can again be used with an 8155 I/O port and control lines. Similar methods can be used with other microprocessors, such as the MC6800 or MCS650X family, and the Intel MCS4/40 family. These both operate almost identically to the method described, except that in the former both R/H and *mode* are tied high, to avoid using a full port for only two lines. Any eight-bit or wider microprocessor (or minicomputer), or narrower devices with eight-bit wide ports (most four-bit devices have eight-bit wide ports available) can be interfaced in a handshake mode with a minimum of additional hardware.

The handshake mode can also be used to interface with industry-standard u.a.r.t.s such as the Intersil IM6402/3 and the Western Digital TR1602. One method is shown in Fig 12. The arrangement here is such that if the u.a.r.t. receives any word serially down the receiver register input line the data received flag will be set. Since this is tied to the *mode* pin, the current result will be loaded, full handshake style as before, into the transmitter buffer register, via the transmitter buffer register empty flag and the t.b.r. *load* lines. The u.a.r.t. will thus transmit the full 18 (16, 14) bit result in three (2, 2) eight-bit words, together with the requisite start, stop and parity bits, serially down the transmitter register output line. If the DR is used to drive R/H, and the received data word used to drive a multiplexer, as shown in Fig. 13, the multiplexer address sent to the u.a.r.t. will be selected, and a conversion initiated of the corresponding analogue input. The result will be returned serially if the *mode* pin is tied high. Thus a complete remote data logging station for up to 256 separate input lines can be controlled and readback through a three line interface. Alternatively, the data word could be used to select one of several A/D converters. The unselected A/Ds all have three stated disable lines as well as data lines, so provided only one device is selected at a time, no conflicts will occur.

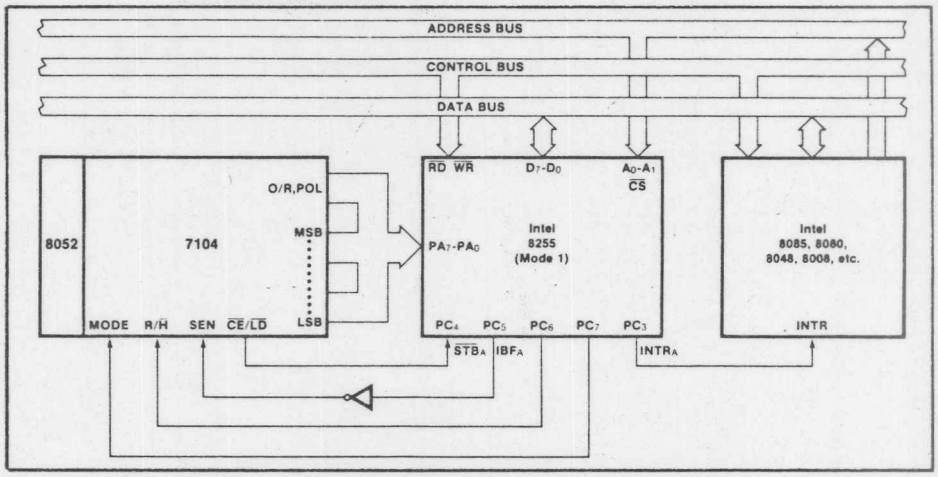


Fig. 11: 8052-A7104 to MCS-48, -80, or 85 handshake interface.

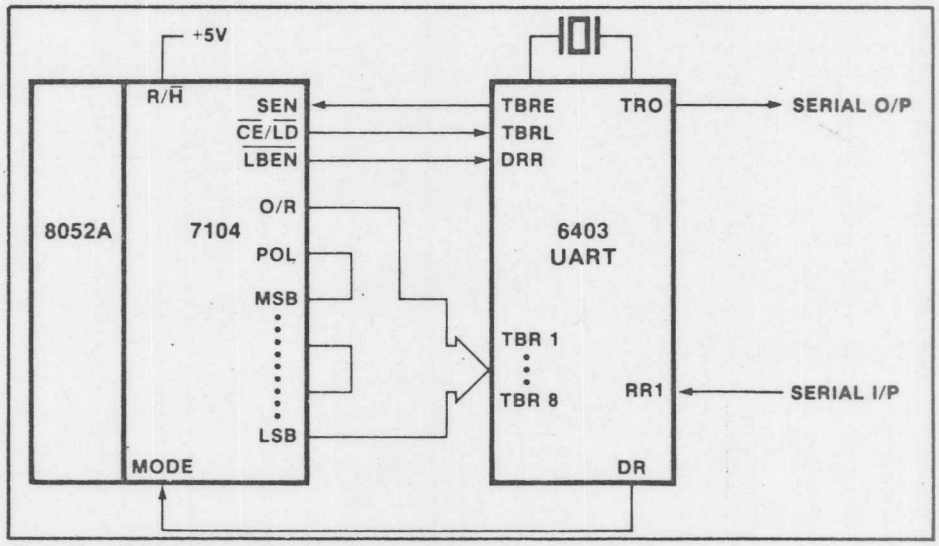


Fig. 12: 8052/7104 serial interface using u.a.r.t.

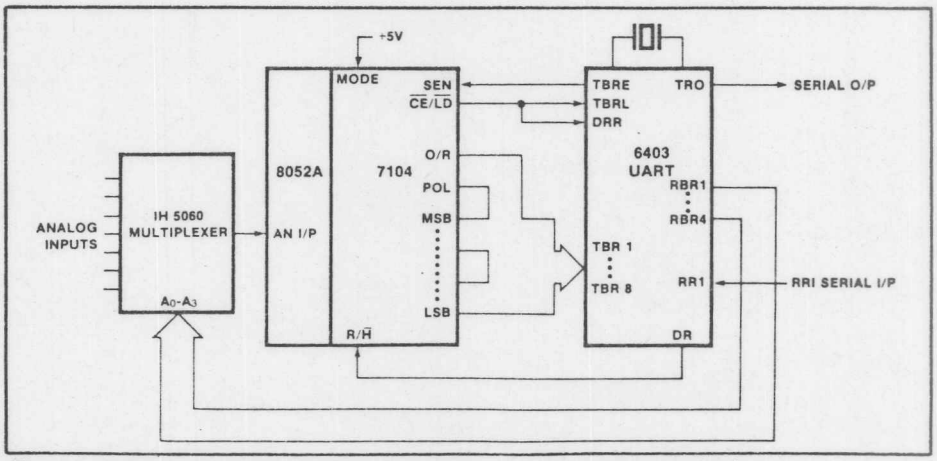
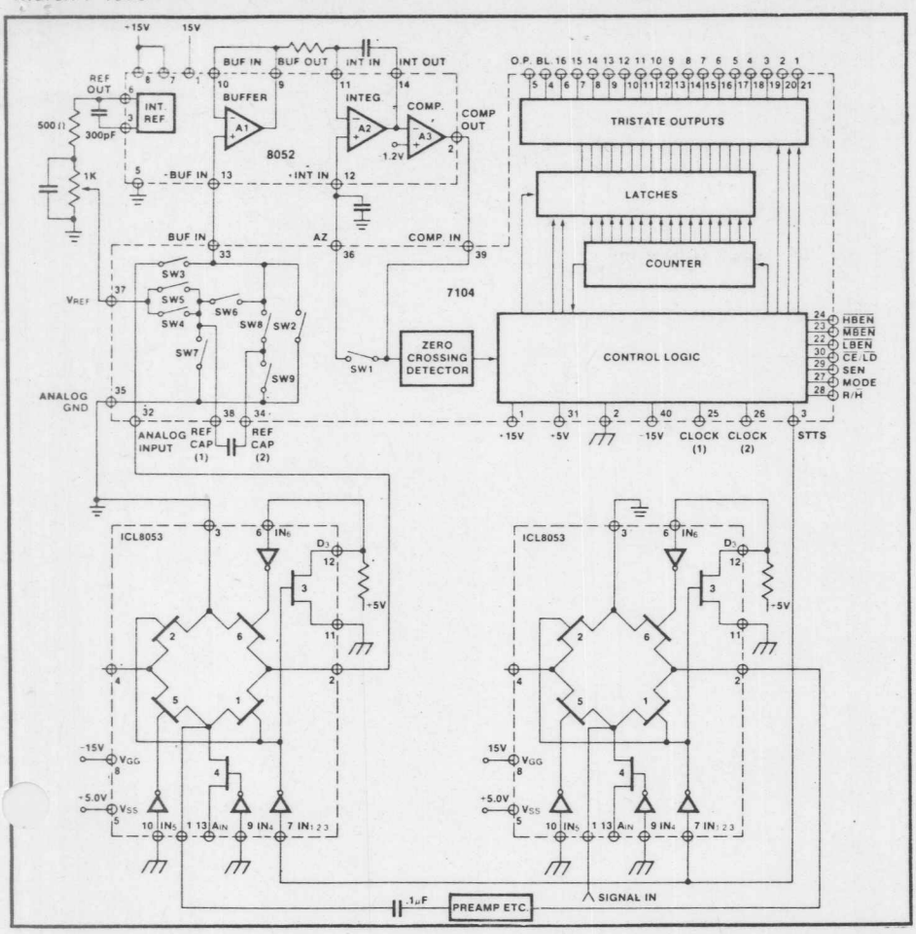


Fig. 13: 8052/7104 serial interface using u.a.r.t. and analogue multiplexer.

Naturally, care must be taken to avoid double selection errors in the data word, or an address decoder used. This technique could also be used to poll many stations on a single set of lines provided that the t.r.o. outputs are either three-state or open collector/drain connections, since only that u.a.r.t. receiving an address which will trigger an attached converter will transmit anything.

A few techniques, primarily analogue, can be used to enhance the performance of the ICL8052/ICL7104 chip pair for certain applications. These are: buffer gain, for sensitivity increases of up to about 5 or 10 to 1; external auto-zero for ▶ *180*



time in auto-zero; i.e., at the maximum possible rate. The necessary "activity" on the R/H pin can be readily provided by tying it to the clock out pin (pin 26). Obviously under these conditions, the conversion cycle time depends on the result.

In many systems, signal conditioning is required in front of the converter for pre-amplification, filtering, etc. With the exception of buffer gain, it is generally not possible to include these conditioning circuits in the auto-zero loop. However, a sample-and-difference circuit keyed to the auto-zero phase can be used to eliminate offset and similar errors in preamplifiers, multiplexers, etc. A suitable circuit for a simple system is shown in Fig 15. The ICL8053 is used as a switch here primarily because of its extremely low charge injection (typically well below 10pC), even though it does limit the analogue swing to ±4V. The use of an IH191 or IH5043 avoids this restriction, but increases the charge injection. Note that this circuit has some sensitivity to stray capacitance at the converter input node. The amplifying or conditioning stages indicated in this circuit must be capable of passing the chopping frequency with small enough delay, rise time, and overshoot to lead to insignificant error. Filtering should be done before or after the switching devices. Note also that although the input signal is still integrated over the normal time period the input reference level is not. The time constant of the hold capacitor charging circuit should take noise and interference effects into consideration.